Georgia Tech Team Entry for the 2011 AUVSI International Aerial Robotics Competition

Girish Chowdhary[*], Daniel Magree[†], Dmitry Bershadsky[†],
Timothy Dyer[‡], Eohan George[§], Hiroyuki Hashimoto[†],
Roshan Kalghatgi[¶], Eric N. Johnson[‖]

Georgia Institute of Technology, Atlanta, GA, 30332-0152, USA

This paper describes the details of a Quadrotor Unmanned Aerial Vehicle capable of exploring cluttered indoor areas without relying on any external navigational aids. An elaborate Simultaneous Localization and Mapping (SLAM) algorithm is used to fuse information from a laser range sensor, an inertial measurement unit, and an altitude sonar to provide relative position, velocity, and attitude information. A wall-following guidance rule is implemented to ensure that the vehicle explores maximum indoor area in a reasonable amount of time. A model reference adaptive control architecture is used to ensure stability and mitigation of uncertainties. The vehicle is intended to be Georgia Tech Aerial Robotic Team's entry for the 2011 International Aerial Robotics Competition.

I. Introduction

The Army, Navy, and the Air Force have identified indoor reconnaissance and surveillance capability as a top research priority due to the changing nature of the battlefield. Miniature air vehicles are ideal candidates for such missions as they can use three dimensional maneuvers to overcome obstacles that cannot be overcome by ground vehicles. However, significant technological challenges exist in order to ensure reliable operation in such environments. Most current algorithms for Unmanned Aerial System (UAS) Guidance Navigation and Control rely heavily on GPS signals,[1, 2, 3] and hence are not suitable for indoor navigation where GPS signal is normally not available. Furthermore, the indoor UAS must be sufficiently small in order to successfully navigate cluttered indoor environments, consequently limiting the amount of computational and sensory power that can be carried onboard the UAS. Finally, the UAS should be designed to be expendable due to the dangerous environments it needs to operate in, hence low-cost, low-weight designs need to be explored. These restrictions pose significant technological challenges for the design of reliable Miniature Air Vehicle (MAV) platforms capable of navigating cluttered areas in a GPS denied environment.

A. Problem Statement

The sixth mission of the International Aerial Robotics competition requires that a MAV weighing less than 1.5kg have the ability to enter and navigate within an unknown confined environment in search of a specific marked target without being detected. The mission also requires that the MAV locate and bring back a flash drive.

[*] Research Engineer II, School of Aerospace Engineering

[†] Graduate Research Assistant, School of Aerospace Engineering

[‡] student, School of Aerospace Engineering

[§] Graduate Research Assistant, School of Electrical and Computer Engineering

[¶] Student, School of Mechanical Engineering

[‖] Lockheed Martin Professor of Avionics Integration, School of Aerospace Engineering

B. Conceptual Solution

The Georgia Tech Aerial Robotics Team (GTAR) has developed an indoor MAV Unmanned Aerial System (UAS) capable of exploring cluttered indoor areas without relying on external navigational aids such as GPS. The MAV uses an off-the-shelf Quadrotor platform and is equipped with off-the-shelf avionics and sensor packages. We use an elaborate navigation algorithm that fuses information from a laser range sensor, inertial measurement unit, and sonar altitude sensor to form accurate estimate of the vehicle attitude, velocity, and position relative to indoor structures. We leverage the fact that all indoor structures have walls to design a guidance algorithm that detects and follows walls to ensure that the navigation solution maintains its fidelity and maximum indoor area is explored in a reasonable amount of time. We use a control architecture that augments a proven baseline proportional-derivative controller with an optional adaptive element that aids in mitigating modeling error and other system uncertainties.

C. Yearly Milestones

For the first competition of the 6th IARC mission, the GTAR Team aims to develop a stable MAV capable of carrying an avionics payload for completing the 6th mission. In 2011, the vehicle will have the ability to navigate and explore the indoor environment, and detect the flash drive. Subsequent yearly milestones include exploring the indoor environment without being detected and a method for picking-up and bringing back the USB disk within allocated time.

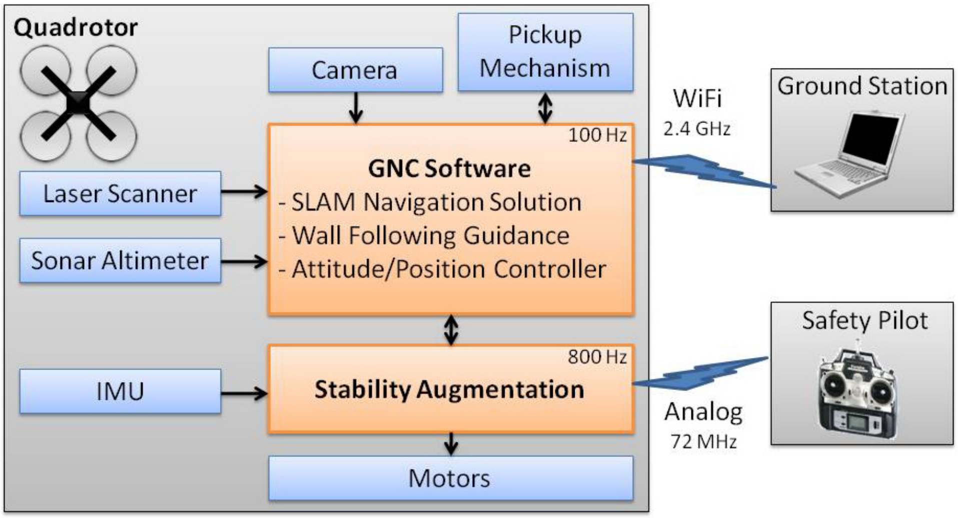


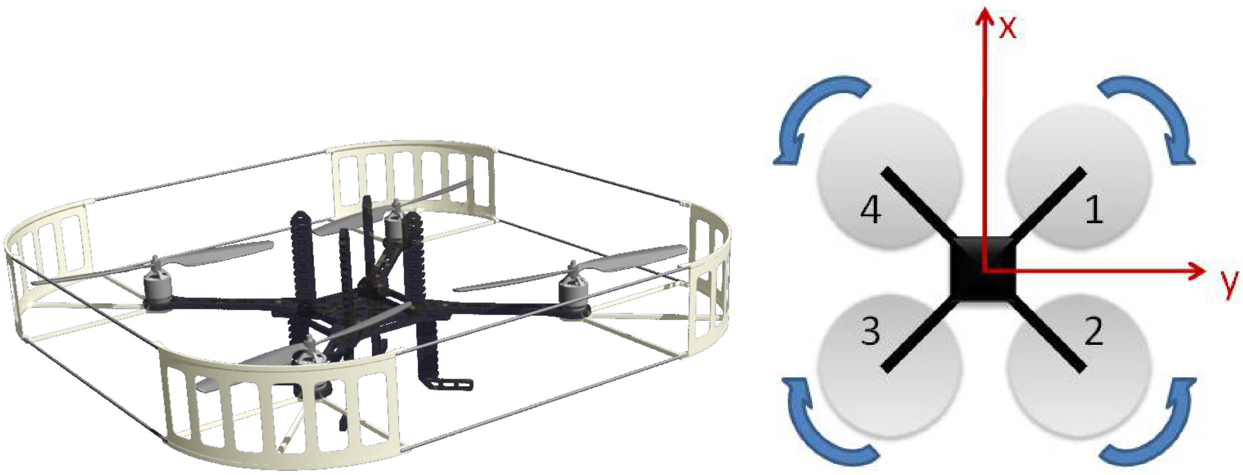
Figure 1. GTAR System Architecture

II. Description of Vehicle

A. Aerial Platform

Quadrotors have become a very popular choice for MAV due to their relatively high payload capacity and high maneuverability.[4] Furthermore, unlike helicopters, Quadrotors avoid the use of mechanical parts for exerting moments and forces required for maneuvering. The 2009 GTAR team used a coaxial helicopter for 2009 IARC competition,[5] however, no off-the-shelf coaxial rotorcraft platforms are available to meet the 2011 payload requirements. Hence, in 2011, we will use a Quadrotor platform.

We selected the AscTec Pelican Quadrotor made by Ascending Technologies GmbH as the base airframe (see Fig.2(a)). The vehicle structure, motors, and rotors of AscTec Pelican were used without modification. The vehicle generates lift using four fixed pitch propellers driven by electric motors. Control is achieved by creating a relative thrust offset between the propellers. Quadrotors can either be flown with diamond configuration (front, right, back, left motors are used to effect pitching, rolling, and yawing motion) or square configuration (front-right, back-right,back-left, front-left motors are used to effect pitching, rolling, and yawing motion). Although many Quadrotors in aerial robotics community fly with diamond configuration,[6,7,4] we selected the square configuration (shown in Fig 2(b)) to allow for more flexibility in sensor mounting locations.



(a) AscTec Pelican Platform

(b) Quadrotor Flying in Square Configuration

Figure 2. The 2011 GTAR Team chose to use the Ascending Technologies Pelican as aerial platform.

B. Guidance Navigation and Control System

1. Guidance System

The developed guidance system uses only locally available information gathered through the onboard sensors, which include a laser range scanner. A simple velocity field method was employed, with a few basic rules for identifying different topology of interest. In essence, the shape of the environment determines where the vehicle decides to go. Each point in a laser scan contributes a small incremental velocity, and the average velocity over all the scan points is used as a commanded velocity to the velocity control loop. Equation 1 gives the equation used to calculate the incremental velocity at each scan angle.[8]

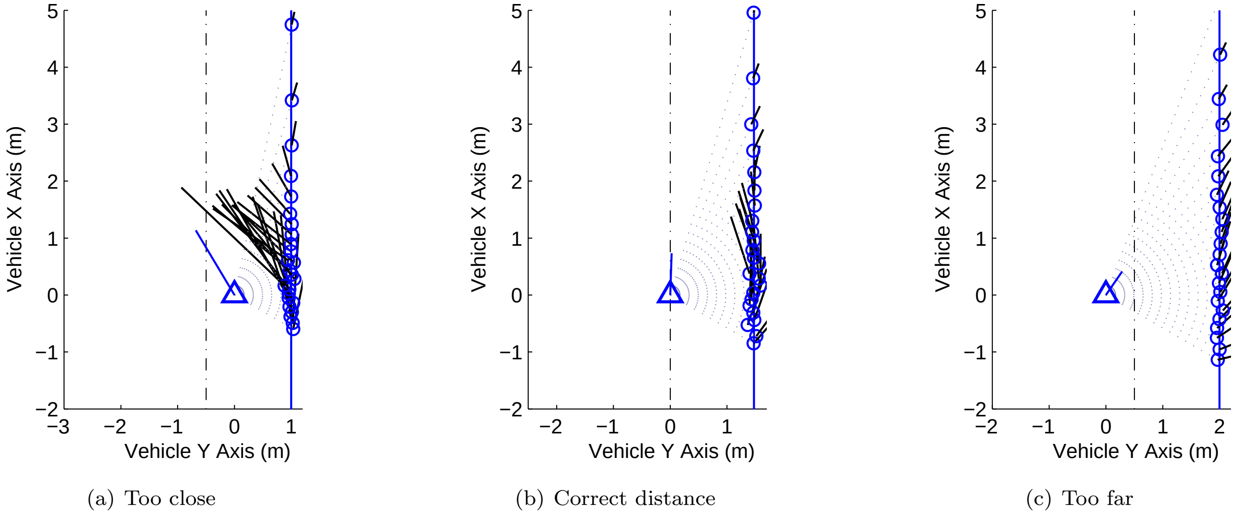
$$\begin{cases} \gamma_i &= \theta_i + \frac{\pi}{2}(\frac{r_i}{r_{goal}}) - \pi \qquad (0 \le r_i \le 2r_{goal}) \\ \gamma_i &= \theta_i \qquad (2r_{goal} < r_i) \end{cases} \tag{1}$$

The incremental velocity at each measurement angle, γ_i, is determined by the range measurement, r_i, the range angle, θ_i and the user-defined desired distance from the wall, r_{goal}. The commanded velocity in the body frame is calculated according to Eq. (2), where K_v is the velocity gain and n is the number of scan points with valid range readings. A user-defined minimum safe distance r_{min} (by default set to the vehicle maximum radius) is used to adaptively increase the gain K_v as a means of providing obstacle avoidance behavior as an inherent part of the velocity guidance algorithm. The body frame velocity commands are transformed to the local inertial frame using the vehicle estimated heading, ψ, as shown in Eq. (3).

$$\begin{cases} u_{cmd_{body}} = \frac{K_v}{n} \sum_{i=1}^{n} \frac{cos(\gamma_i)}{|r_k - r_{min}|} \\ v_{cmd_{body}} = \frac{K_v}{n} \sum_{i=1}^{n} \frac{sin(\gamma_i)}{|r_k - r_{min}|} \end{cases} \tag{2}$$

$$\begin{bmatrix} u_{cmd} \\ v_{cmd} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} u_{cmd_{body}} \\ v_{cmd_{body}} \end{bmatrix} \tag{3}$$

Heading control is achieved by using the angle parameters of the line segments extracted from the scan data using the Polar Recursive Least Squares (PRLS) algorithm. The vehicle is given a commanded heading to align itself with a portion of the wall. The default behavior is to use the farthest contiguous wall segment, starting from the beginning of the scan on the right and continuing counterclockwise until a gap is found in the wall that is large enough for the vehicle to fly through. Using this setting causes the vehicle to turn around when it encounters a "dead end" hallway while still making proper outside turns to go through doors or into side hallways. The average of the incremental velocity commands provide the vehicle with a flight direction and speed that varies appropriately depending on the local environment topology. This guidance algorithm produces basic wall-following behavior, whereby the vehicle will continue along a wall until it encounters an opening, an inside corner, or an outside corner. As the vehicle is flying along a wall, small changes in the velocity command keep the vehicle at the desired distance from the wall as shown in Fig. 3. The algorithm also mitigates inside and outside corners (see reference 9 for terminology description) as shown in figure 4. The default behaviour of detecting and mitigating gaps in the environment is shown in 5. Further information can be found in reference 8 and 10.

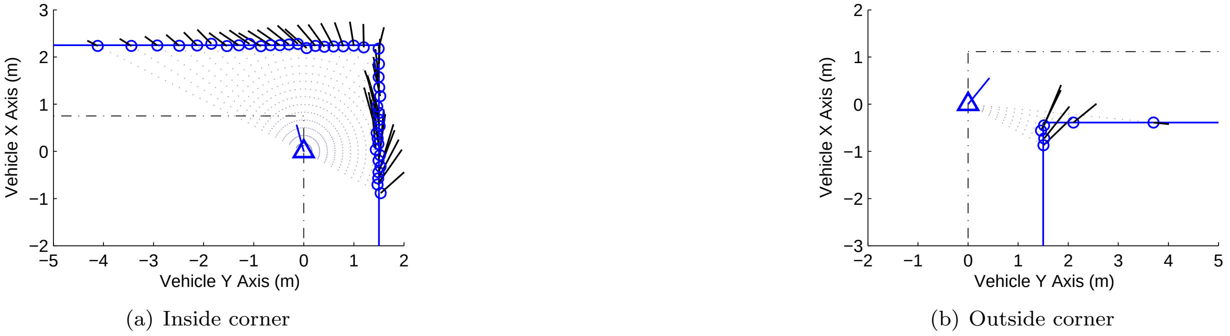


(a) Too close

(b) Correct distance

(c) Too far

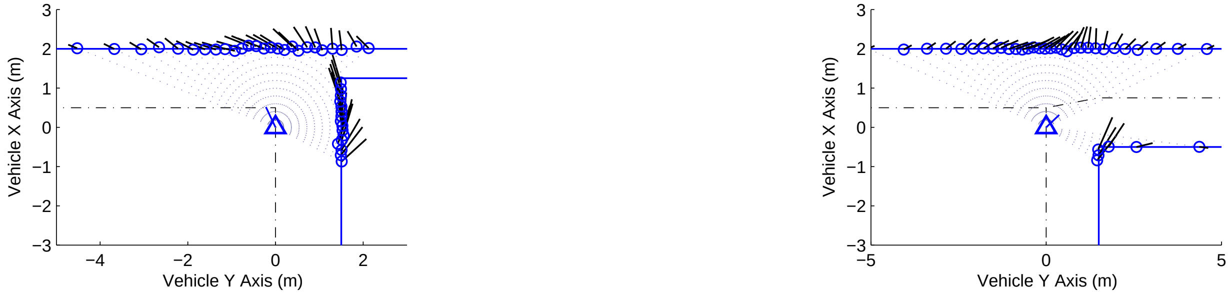
Figure 3. The guidance algorithm generates a velocity vector from the average of vectors generated for each range measurement. Along a section of straight wall, the algorithm gives corrections to keep the vehicle a desired distance from the wall, which is indicated by a dotted line. The wall is shown by a solid vertical line, with individual range measurements shown by circles. Individual velocity increments are indicated by a line showing the magnitude and direction at each measurement point. The total commanded velocity is shown by a line emanating from the vehicle, which is represented by a triangle.



(a) Inside corner

(b) Outside corner

Figure 4. Inside and outside corners are handled with no special topology check required. The average velocity vectors generate a total commanded velocity that takes the vehicle smoothly through the maneuver. Heading commands are generated to turn the vehicle parallel with the next wall face as soon as it is detected.



(a) Small gap

(b) Large Gap

Figure 5. Any gap in the wall is detected by the PRLS wall estimation algorithm. Gaps that are smaller than the desired threshold are flown by with no effect on the commanded velocity. Gaps that are large enough for the vehicle to fly through, such as a hallway or door, cause the commanded velocity vector to bend toward the opening. If the vehicle enters a hallway or goes through a door where the minimum lateral separation is not at least twice the wall follow distance, the vehicle simply keeps an equal distance from both sides. Otherwise, the vehicle picks up the wall on the right hand side and keeps the desired distance from it as flight continues.

2. Stability Augmentation System (SAS)

The Quadrotor platform is inherently unstable, that is, without control inputs, the platform would enter an uncontrolled drift in velocity and angular rates and collide with the ground or nearby obstacles. Quadrotors are also known to be notoriously hard to control even for human pilots, particularly because the relationship between thrust and stick deflection is nonlinear and because attitude is coupled heavily with velocity. Hence, it is desirable to integrate angular rate damping to aid the pilot in controlling the Quadrotor. Let $\hat{p}$, $\hat{q}$, and $\hat{r}$ denote the gyroscope measurements of the Quadrotor roll, pitch, and yaw rates, and δ_{ϕ_p}, $\delta_{\theta p}$, and δ_{ψ_p} denote the pilot roll, pitch, and yaw stick deflections, then the actual stick deflection commands are assigned using the following proportional control logic:

$$\delta_\phi = \delta_{\phi_p} - K_p \hat{p}, \tag{4}$$

$$\delta_\theta = \delta_{\theta p} - K_q \hat{q}, \tag{5}$$

$$\delta_\psi = \delta_{\psi_p} - K_r \hat{r}. \tag{6}$$

In equation 4, K_p, K_q, and K_r denote the linear gains chosen to provide appropriate rate damping.

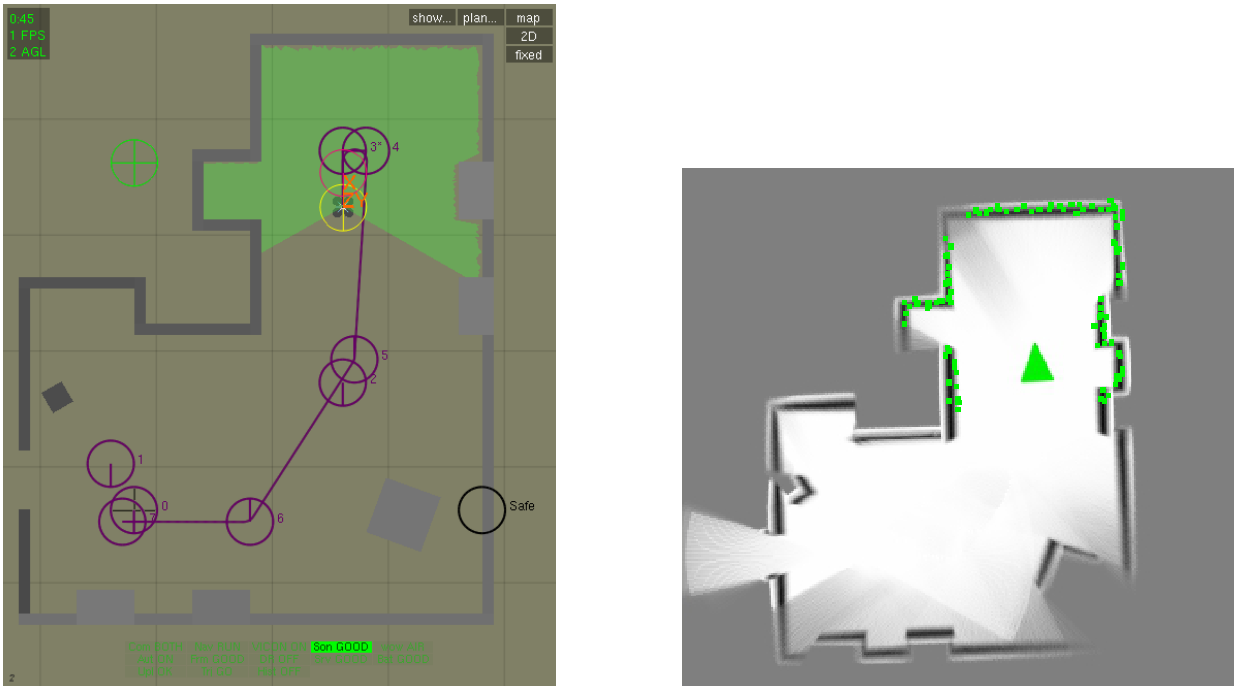
3. Control Algorithm

The complexity of the control system depends not only on the quantities being controlled, but also on the dynamics of the system itself. Unlike ground vehicles, unstable air vehicles are susceptible to oscillation and divergent flight when the control system is not properly tuned. Even for stable flying vehicles, coupling between lateral and longitudinal motion as well as aerodynamic interaction with the environment must be considered. The control architecture used by the GTAR 2011 team leverages the proven Model Reference Adaptive Control architecture developed for control of VTOL UAS throughout their flight envelop by Georgia Tech UAV Research Facility.[11,12,13] In this architecture, a position control loop generates a velocity command, a velocity control loop generates an attitude command, and an attitude control loop generates servo commands to stabilize the vehicle by controlling the angular rate. Kannan has shown that such nested and cascaded control loop architecture with actuator saturation can indeed be used to control VTOL UAS.[12] This system of nested control loops requires that the vehicle maintain an estimate of its position, velocity, attitude, and angular rate. For details the reader is referred to references 14, 11, 12, 15 for further details.

4. Navigation Algorithm

A variety of SLAM algorithm implementations are available for free use at the web site OpenSLAM.org. The algorithm used for the preliminary research, called CoreSLAM,[16] was chosen primarily because it is simple, easy to implement, and it uses integer math where possible to improve computational speed.[17] There are two main parts to any SLAM routine. The first task is to measure distance to obstacles or landmarks in the environment, and to map them given the vehicle's position and orientation (i.e. mapping). The second task is to determine the best estimate of the vehicle's position and orientation based on the latest scan (or series of scans) given a stored map (i.e. localization). The mapping and localization tasks are performed together to maintain the most current map and position estimate. The GTAR team has developed an Iterative Closest Point (ICP) scan-matching algorithm in-house. The algorithm were developed to prioritize localization with respect to the immediate environment, and place lesser emphasis on building and maintaining highly accurate global maps. Hence the "map" in the ICP algorithm consists entirely of a singe previous scan. However, with the ICP algorithm individual scans can be saved and post-processed into a global map if desired.

Figure 6(a) shows a simulated building interior being explored by the Quadrotor with a scanning laser rangefinder. Figure 6(b) shows the map generated during a simulated flight.



(a) Mapping a simulated environment

(b) Map generated during simulation

Figure 6. The map maintained by the CoreSLAM routine represents an occupancy grid, where the value of each pixel in the image represents the likelihood that a particular grid square is occupied. Here, lighter colors represent free space, while darker colors represent obstacles and medium gray areas are unexplored. Areas with higher contrast represent greater certainty due to longer observation periods during the flight. The green triangle represents the vehicle's estimated position and heading.

The inherent nonlinearities in the vehicle dynamics and the measurement sensors are handled through the use of an Extended Kalman Filter (EKF). An existing EKF-based navigation filter architecture developed at the Georgia Tech UAV Research Facility is utilized as part of the indoor navigation system.[2] The navigation algorithm developed by the GTAR team augments the existing EKF based architecture to function without GPS signals by using the range information obtained from the laser range scanner and the sonar altimeter.[10]

C. Memory Stick Retrieval

Once the target memory stick is located, a specialized guidance system generates a trajectory for the vehicle for landing on and retrieving the memory stick. The guidance system estimates the three-dimensional location of the memory stick using a downward-facing camera mounted on the vehicle. A cascade filter identifies the pixel location and pixel area of the memory stick in each image from the camera, and an EKF estimates the three-dimensional location based on the pixel measurements. The vehicle is then commanded to decend onto the location of the memory stick and retrieve it. One preliminary retrieval mechanism design is a pad of adhesive tape on the bottom of the vehicle.

Let the camera frame coordinates of the memory stick be given by $\bar{x_c} = [XYZ]^T$. Measurements from the camera are the square-root of the pixel area, A_{sqrt_p}, and pixel location, x_p and y_p, and define measurement vector $\bar{y} = [A_{sqrt_p} x_p y_p]^T$.The equations for the measurement model are given by

$$A_{sqrt_p} = \frac{k}{X}\sqrt{A_{ft}} \tag{7}$$

$$x_p = \frac{k}{X}Y \tag{8}$$

$$y_p = \frac{k}{X}Z \tag{9}$$

where A_{ft} is the area of the memory stick in feet and k is the pixel to radian scale factor, both known constants. The linearized measurement model is generated by taking the first order Taylor series expansion of the previous equations.

$$\frac{\partial \bar{y}}{\partial \bar{x_c}} = \begin{bmatrix} -\frac{k}{X^2}\sqrt{A_{ft}} & 0 & 0 \\ -\frac{k}{X^2}Y & \frac{k}{X} & 0 \\ -\frac{k}{X^2}Z & 0 & \frac{k}{X} \end{bmatrix} \tag{10}$$

Converting to the inertial frame,

$$\frac{\partial \bar{y}}{\partial \bar{x}} = \frac{\partial \bar{y}}{\partial \bar{x_c}}\frac{\partial \bar{x_c}}{\partial \bar{x}} = \frac{\partial \bar{y}}{\partial \bar{x_c}}L_{ci} = \mathbf{C} \tag{11}$$

where L_{ci} is the rotation matrix converting from the cameraframe to the inertial frame. Inserting matrix $\mathbf{C}$ into the EKF measurement update gives the corrected state.

$$\mathbf{K} = \mathbf{P}^- \mathbf{C}^T(\mathbf{C}\mathbf{P}^- \mathbf{C}^T + \mathbf{R})^{-1} \tag{12}$$

$$\hat{x} = \hat{x}^- + \mathbf{K}[\bar{y} - h(\hat{x}^-)] \tag{13}$$

$$\mathbf{P} = (\mathbf{I} - \mathbf{K}\mathbf{C})\mathbf{P}^- \tag{14}$$

where $\mathbf{R}$ is the measurement covariance matrix, $\hat{x}$ is the inertial frame measurement estimate, and $h(\hat{x}^-)$ is the output of the nonlinear measurement model. This system estimates the three-dimensional location of the memory stick with sufficient accuracy to be able to land the aircraft on the target.

D. Flight Termination System

A manual takeover switch is provided so that a human safety pilot can take over control when required. The system is also provided with a remotely controlled “kill-switch” that ensures power to the motors is killed when triggered.

III. Payload

A. Sensor Suite

The GTAR entry uses three primary measurement sensors for navigation, stability and control. These devices are a laser range finder and a sonar altimeter for the GNC software, and an inertial measurement unit (IMU) for stability augmentation. The IMU employed by the GTAR team is the ADIS-16365-BMLZ built by Analog Devices Inc. It consists of a tri-axis digital gyroscope and tri-axis accelerometere that can measure forces up to ±18 g. The laser range finder used is the Hokuyo URG-04LX-UG01. It is capable of measuring distances up to 4 m and has a maximum detection area of 240 degrees, with a resolution of 1 mm and 0.36 degrees respectively. The sonar altimeter used is the MB1040 LV MaxSonar EZ4 high performance ultrasonic range finder. It is capable of measuring distances up to 6.45 m away with resolution of 25.4 mm. These sensors are integrated around the Gumstix Overo Fire onboard computer which is a small and cost effective ARM Cortex-A8 OMAP3530 based computer-on-module. It is equipped with 256 MB Flash RAM, I2C and is UART and SPI capable. The onboard software leverages our previous work in rotorcraft control and allows us to fully utilize the GUST software suite of the GeorgiaTech UAV Research Facility.[18]

B. Communications

The Gumstix overo computer can communicate via 802.11g and Bluetooth wireless links. The computer will communicate with a ground computer using a wireless Local Area Network (LAN) link.

C. Power Management system

The system uses off-the-shelf battery packs which have a track record of proven safety. Two three cell Lithium Polymer Ion batteries are used, one drives the motors to provide lifting power, and the other runs the onboard computer. The batteries are charged off-board using off-the-shelf battery chargers. The vehicle comes with an integrated audible low-power warning, battery voltage can also be measured and transmitted to the ground station.

IV. Operations

A. Flight Preparations

Before each autonomous flight test or competition trial, a checklist of preparations are to be followed (see table A).

Table 1. Flight Checklist

Steps completed days before flight session	Charge flight batteries, transmitter batteries Load new software onboard and ground station Complete hardware-in-the-loop (HITL) tests to ensure proper operation of any code changes
Steps completed day of flight session	Ensure all flight test equipment is present. Set up ground station.
Steps completed before each flight	Clearly brief safety pilot of intention of flight Check structural integrity of vehicle and ensure proper center-of-gravity position.
During flight test	Pilot has primary discretion on whether to take manual control if vehicle is in jeopardy. Besides this discretion, safety pilot will only obey judges or ground station operator. Once the low voltage warning tone is heard, safety pilot takes control and lands the aircraft.

B. Man/Machine Interface

A ground station based on the GIT GUST software environment will continuously monitor the flight vehicle and display health and status information during the flight.[18] The flight vehicle will send its current estimated position/heading, obstacle locations, and battery voltage via a wireless LAN data link. In addition, a frame-grabber is used to retrieve images from the incoming video stream for processing. Instructions from the ground station, including the adjustment of system parameters during manual flight, are transmitted over a wireless LAN data link. A safety pilot link is included, which operates via a separate 2.4GHz radio uplink.

V. Risk Reduction

A. Vehicle Status Monitoring

The flight vehicle continually monitors its surroundings for potential hazards and obstacles using the onboard laser range scanner. The information about potential hazard can be transmitted to the ground station for monitoring.

1. *Shock and vibration isolation*

The chosen onboard electronics have inherent tolerance to shock and vibration. Further vibration reduction is achieved through careful mounting of the hardware. The avionics package is mounted close to the center of gravity to minimize motion induced due to body rotations. The IMU is mounted directly on the avionics board. The laser range scanner and the sonar altimeter are mounted using a low cost vibration isolation mechanism.

2. *Electromagnetic Interference (EMI)/Radio Frequency Interference (RFI) Solutions*

The chosen quadrotor platform has brushless motors, which has reduced EMI signature. Further EMI mitigation is achieved by mounting the avionics package at the center of the airframe, and thus spatially separating it from the motors. Proper electric grounding and additional capacitors are used to provide further protection against EMI. A 2.4GHz transmitter was chosen for the video link, the safety pilot radio control link, and the data link. This eliminates the typical "servo jitter"affecting UAVs operating with 900MHz transmitters nearby. Possible interference between the different 2.4GHz systems is reduced by proper shielding and location of antennas.

B. Safety to Bystanders

The Quadrotor platform used in this work has a protective shroud that minimizes the risk of rotor strike and improves crash-worthiness. Further safety is incorporated by using off-the-shelf battery packs with a track record of proven safety. A manual takeover switch is provided so that a human safety pilot can take over control when required. Finally, the system is also provided with a remotely controlled "kill-switch" that ensures power to the motors is killed when triggered.

C. Modeling and Simulation

1. *Modeling*

Quadrotor dynamics flying in configuration shown in Fig.2(b) has been modeled in simulation. Assuming near hover aerodynamics, fuselage aerodynamics and forward flight rotor aerodynamics can be neglected. Hence, the total force acting on the Quadrotor is composed only of thrust and gravity forces. Newton's second law in the body axis can be written as:

$$\begin{bmatrix} 0 & 0 & -(\tau_1+\tau_2+\tau_3+\tau_4) \end{bmatrix}^T + F_g = m\frac{{}^b dv}{dt} + \omega \times mv \tag{15}$$

where τ_i represents thrust magnitude on the ith rotor for $i = 1, 2, 3, 4$. F_g represents gravity force acting on the vehicle in the body frame, $v \in \Re^3$ is the velocity in the body frame, $\frac{{}^b dv}{dt}$ is the derivative of the body velocity with respect to the body frame, and $\omega \in \Re^3$ is the angular rate of the body.

Neglecting forward flight aerodynamics, total moment acting on the Quadrotor composes of four different sources: hub yawing moment (M_{hy}), differential thrust moment (M_{dt}), inertial reaction moment (M_{ir}), and gyroscopic moment (M_{gy}).[4,6,7] The primary moment contributions are from hub yawing moment and differential thrust moment while inertial reaction moment and gyroscopic moment provides insignificant contribution. Euler's law in the body axis can be written as

$$M = M_{hy} + M_{dt} + M_{ir} + M_{gy} = I\dot{\omega} + \omega \times I\omega \tag{16}$$

where I is the Quadrotor inertia matrix. The individual moment components are defined as follows:

$$M_{hy} = \begin{bmatrix} 0 & 0 & -M_1+M_2-M_3+M_4 \end{bmatrix}^T \tag{17}$$

$$M_{dt} = \begin{bmatrix} l_y(-\tau_1-\tau_2+\tau_3+\tau_4) & l_x(\tau_1-\tau_2-\tau_3+\tau_4) & 0 \end{bmatrix}^T \tag{18}$$

$$M_{ir} = \begin{bmatrix} 0 & 0 & -(I_r\dot{\Omega}_1 - I_r\dot{\Omega}_2 + I_r\dot{\Omega}_3 - I_r\dot{\Omega}_4) \end{bmatrix}^T \tag{19}$$

$$M_{gy} = \begin{bmatrix} -q(I_r\Omega_1 - I_r\Omega_2 + I_r\Omega_3 - I_r\Omega_4) & p(I_r\Omega_1 - I_r\Omega_2 + I_r\Omega_3 - I_r\Omega_4) & 0 \end{bmatrix}^T. \tag{20}$$

where M_i is hub yawing moment magnitude of the i^{th} rotor acting on the body. Ω_i is angular velocity magnitude of the i^{th} rotor. I_r is moment of inertia of the rotor blade. l_x and l_y are distance from center of gravity to rotor hub in x and y direction. p and q are roll and pitch rates written in body axis. The total moment acting on the body is formed by combining the above terms:

$$M = \begin{bmatrix} l_y(-\tau_1 - \tau_2 + \tau_3 + \tau_4) - q(I_r\Omega_1 - I_r\Omega_2 + I_r\Omega_3 - I_r\Omega_4) \\ l_x(\tau_1 - \tau_2 - \tau_3 + \tau_4) + p(I_r\Omega_1 - I_r\Omega_2 + I_r\Omega_3 - I_r\Omega_4) \\ -M_1 + M_2 - M_3 + M_4 - (I_r\dot{\Omega}_1 - I_r\dot{\Omega}_2 + I_r\dot{\Omega}_3 - I_r\dot{\Omega}_4) \end{bmatrix}. \tag{21}$$

Vehicle's pitching motion can be generated by commanding differential Revolutions Per Minute (RPM) between two front motors and two back motors. Rolling motion can be generated by commanding differential RPM between two right motors and two left motors. Yawing motion can be generated by increasing motor RPMs on one diagonal and reducing motor RPMs on the other diagonal.[6] Typical helicopter controls can therefore be mapped to motor commands x_i using Eq.(22).

$$\begin{aligned} x_1 &= \sqrt{\delta_T - \delta_\phi + \delta_\theta + \delta_\psi} & x_2 &= \sqrt{\delta_T - \delta_\phi - \delta_\theta + \delta_\psi} \\ x_3 &= \sqrt{\delta_T + \delta_\phi - \delta_\theta - \delta_\psi} & x_4 &= \sqrt{\delta_T + \delta_\phi + \delta_\theta + \delta_\psi} \end{aligned} \tag{22}$$

where δ_T, δ_ϕ, δ_θ, and δ_ψ represent thrust, roll, pitch, and yaw commands. In this form the commands are equivalent to a helicopter's collective, lateral cyclic, longitudinal cyclic, and tail rotor commands. A second order model is used to relate motor command x_i to motor RPM Ω_i. The second order model consist of two cascaded first order systems: the first one relates the motor commands to the motor states, and the second one relates the motor states to rotor RPM. Rotor aerodynamics is modeled using blade element and momentum theory. The relationship between motor command x_i , rotor angular velocity Ω_i, thrust τ_i, and moment M_i is modeled similar to reference 19. Vehicle's parameters including moment of inertias and maximum achievable RPM are currently being refined through experimentation.

2. Simulation

The GTAR team utilizes existing simulation software developed for research projects at Georgia Tech UAV lab.[20] The simulation significantly reduces development time as the team can adopt navigation filter and controller already implemented in other UAVs. The simulation comes complete with modeling of uncertainties such as gusts, and modeling of indoor environments. All sensors are elaborately emulated and their noise properties are reproduced for testing purposes. Onboard code developed in simulation is directly used for autonomous flight. The setup is also capable of hardware-in-the-loop (HITL) test (simulating only vehicle dynamics and sensor readings). Figure 2 depicts a screen-shot of the vehicle in simulation.

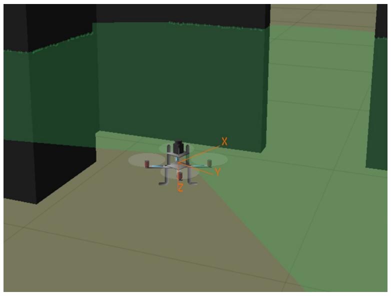


Figure 7. Quadrotor Performing Wall Following in Simulation

VI. Testing

The GTAR system is being rigorously subjected to flight testing at the indoor test flight facility at Georgia Tech. The VICON camera based object tracking system is being used to validate the navigation algorithm. We are using the protocols developed by the GTAR team for the 2009 effort for ensuring safety, efficiency, and reliability in flight tests.

VII. Conclusion

We presented the details of a Quadrotor Unmanned Aerial Vehicle intended for exploring indoor areas. The vehicle uses an off-the-shelf platform equipped with off-the-shelf avionics and sensor packages. Information from a scanning laser range sensor, inertial measurement unit, and a altitude measurement sonar are fused to form an elaborate navigation solution using Simultaneous Localization and Mapping (SLAM) methods. An important feature of this navigation architecture is that it does not rely on any external navigational aid, such as Global Positioning System signal. The information from the navigation solution is processed by a guidance logic which detects and follows walls in an indoor environment to ensure that maximum area of the indoor environment is traversed in a reasonable amount of time.

A cascaded inner-outer loop controller architecture which relates stick commands to attitude commands and attitude commands to velocity commands is used. The control architecture employs an optional adaptive element which can be used to mitigate modeling error and other system uncertainties. The control system also uses a linear Stability Augmentation System that uses rate feedback to dampen the vehicle angular rate response.

An elaborate simulation model of the vehicle has been developed and the navigation and control algorithms have already been validated in simulation. Efforts for flight testing of the vehicle are currently in progress. The Georgia Tech Aerial Robotics team intends to compete in the 2011 IARC competition with this vehicle.

Acknowledgments

The Georgia Tech Aerial Robotics team wishes to thank Jeong Hur, Dr. Suresh Kannan, Nimrod Rooz, and Dr. Erwan Salaün for valuable comments.

References

[1] Kayton, M. and Fried, W. R., *Avionics Navigation Systems*, John Wiley and Sons, 1997.

[2] Christophersen, H. B., Pickell, W. R., Neidoefer, J. C., Koller, A. A., Kannan, S. K., and Johnson, E. N., "A compact Guidance, Navigation, and Control System for Unmanned Aerial Vehicles," *Journal of Aerospace Computing, Information, and Communication*, Vol. 3, May 2006.

[3] Wendel, J., Maier, A., Metzger, J., and Trommer, G. F., "Comparison of Extended and Sigma-Point Kalman Filters for Tightly Coupled GPS/INS Integration," *AIAA Guidance Navigation and Control Conference*, San Francisco, CA, 2005.

[4] Bouabdallah, S., Noth, A., and R., S., "PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor," *Proc. of The IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[5] Chowdhary, G. et al., "Georgia Tech Aerial Robotics Team 2009 International Aerial Robotics Competition Entry," *Proc. of the 1st Symposium on Indoor Flight, International Aerial Robotics Competition*, 2009.

[6] Portlock, J. and Cubero, S., "Dynamics and Control of a VTOL quad-thrustaerial robot," *Mechatronics and Machine Vision in Practice*, edited by J. Billingsley and R. Bradbeer, 2008.

[7] Guo, W. and Horn, J., "Modeling and simulation for the development of a quad-rotor UAV capable of indoor flight," *Modeling and Simulation Technologies Conference and Exhibit*, 2006.

[8] Yamaura, S., "Development of Guidance Algorithm for Unmanned Indoor Flight Vehicle by Using 2D Laser Scanner," Unpublished.

[9] Chowdhary, G., Salaün, E., Ottander, J., and Johnson, E., "Low Cost Guidance, Navigation, and Control Solutions for Miniature Air Vehicle in GPS Denied Environments," *First Symposium on Indoor Flight, International Aerial Robotics Competition*, Mayaguez, Puerto Rico, 2009.

[10] Sobers, M. J., *INDOOR NAVIGATION FOR UNMANNED AERIAL VEHICLES USING RANGE SENSORS*, Ph.D. thesis, Georgia Institute of Technology, 2010.

[11] Johnson, E. and Kannan, S., "Adaptive Trajectory Control for Autonomous Helicopters," *Journal of Guidance Control and Dynamics*, Vol. 28, No. 3, May 2005, pp. 524–538.

[12] Kannan, S. K., *Adaptive Control of Systems in Cascade with Saturation*, Ph.D. thesis, Georgia Institute of Technology, Atlanta Ga, 2005.

[13] Chowdhary, G. and Johnson, E., "Flight Test Validation of Long Term Learning Adaptive Flight Controller," *Proceedings of the AIAA GNC Conference, held at Chicago, IL*, 2009.

[14] Johnson, E. N., *Limited Authority Adaptive Flight Control*, Ph.D. thesis, Georgia Institute of Technology, Atlanta Ga, 2000.

[15] Chowdhary, G. V. and Johnson, E. N., "Theory and Flight Test Validation of Long Term Learning Adaptive Flight Controller," *Proceedings of the AIAA Guidance Navigation and Control Conference*, Honolulu, HI, 2008.

[16] Steux, B. and Hamzaoui, O. E., "Coreslam on openslam.org," website.

[17] Steux, B. and Hamzaoui, O. E., "CoreSLAM : a SLAM Algorithm in less than 200 lines of C code," Tech. rep., Mines ParisTech, Center for Robotics, Paris, France, 2009.

[18] Johnson, Eric N, C. A. J., Watanabe, Y., Ha, J.-C., and Neidhoefer, J., "Vision Only Control and Guidance of Aircraft," *Journal of Aerospace Computing, Information, and Communication*, Vol. 23, No. 10, October 2006.

[19] Johnson, E. and Turbe, M., "Modeling, Control, and Flight Testing of a Small Ducted Fan Aircraft," *Journal of Guidance Control and Dynamics*, Vol. 29, No. 4, July/August 2006, pp. 769–779.

[20] Johnson, E. N. and Schrage, D. P., "System Integration and Operation of a Research Unmanned Aerial Vehicle," *AIAA Journal of Aerospace Computing, Information and Communication*, Vol. 1, No. 1, Jan 2004, pp. 5–18.